

Appl. No. : 09/818,134
 Filed : March 27, 2001

AMENDMENTS TO THE CLAIMS

Claims 1-34 were pending at the time of the January 11, 2005 Office Action. As of this Response, Applicant has amended Claims 25-26, 28, and 31-32. Claims 1-24, 27, 29-30, and 33-34 remain as originally filed.

1. (Original) A method for creating, in response to only a single action by a user, a self-extracting file, the method comprising:

receiving, from a user, an input file to be used in creating a self-extracting file;

and

without further action by the user, creating a self-extracting file using the input file, wherein the input file is automatically launched upon execution of the self-extracting file.

2. (Original) The method of Claim 1, wherein the received input file has an associated filename and wherein a filename for the self-extracting file is automatically generated based in part on the associated filename of the received input file.

3. (Original) A method for creating, in response to a single action, a self-extracting file from an associated input file, wherein the associated input file is automatically launched upon execution of the self-extracting file, and wherein a user is not required to separately choose a data compression method, create a compressed archive using the chosen compression method, select an input file to be launched upon decompression of the compressed archive, and create a self-extracting file from the compressed archive, the method comprising:

receiving an input file to be used in creating a self-extracting file, wherein the file may be of any file type; and

in response to only a single action, creating a self-extracting file from the input file, wherein the input file is automatically launched upon execution of the self-extracting file.

4. (Original) The method of Claim 3, wherein the single action is a single click.

5. (Original) The method of Claim 3, wherein the single action is a double click.

6. (Original) The method of Claim 3, wherein the single action is speaking a sound.

7. (Original) The method of Claim 3, wherein the single action is pressing a key.

Appl. No. : 09/818,134
 Filed : March 27, 2001

8. (Original) The method of Claim 3, wherein the single action is a call from a software routine.

9. (Original) The method of Claim 3, further comprising generating a filename for the self-extracting file, wherein the generated filename is based on a filename associated with the input file.

10. (Original) A method for creating a self-extracting file, the method comprising:
 receiving an input file to be used in creating a self-extracting file, wherein the input file is of any file type; and
 automatically creating a self-extracting file using the received input file.

11. (Original) The method of Claim 10, wherein the creation of the self-extracting file comprises:

opening an output file;
 attaching a decompression engine to the output file, wherein the decompression engine is capable of decompressing compressed data to a temporary file;
 attaching a loader to the output file, wherein the loader configures the output file so as to automatically launch the temporary file after execution of the self-extracting file;
 compressing the received input file according to a data compression method;
 attaching an archive header including information about the compressed input file;
 and
 closing the output file, wherein the closed output file is the self-extracting file.

12. (Original) The method of Claim 11, wherein the input file is received from a computer user.

13. (Original) The method of Claim 11, wherein the input file is received from a software routine.

14. (Original) The method of Claim 11, wherein the data compression method is the same method for all received input files.

15. (Original) The method of Claim 11, wherein the data compression method is determined based on the file type of the received input file.

16. (Original) The method of Claim 11, wherein the loader attached to the output file depends on the file type of the input file.

Appl. No. : 09/818,134
 Filed : March 27, 2001

17. (Original) The method of Claim 11, wherein the loader automatically unloads the temporary file.

18. (Original) The method of Claim 11, further comprising attaching an unloader to the output file to automatically unload the temporary file.

19. (Original) The method of Claim 18, wherein the unloader performs cleanup processes on the temporary file.

20. (Original) A method for creating an executable file, comprising, in response to only a single action, creating a self-extracting file from any input file, wherein the input file may be of any file type, and wherein the input file will be automatically launched upon execution of the self-extracting file.

21. (Original) A method of creating a self-extracting file comprising:

displaying a first frame allowing a user to specify an input file to be converted to a self-extracting file;

receiving the input file specified by the user, wherein the received input file is automatically configured as a self-extracting file and wherein the input file is automatically launched upon execution of the self-extracting file; and

displaying a second frame, wherein the second frame includes a link to the self-extracting file created from the user specified input file.

22. (Original) A system for creating a self-extracting file comprising:

a receiving module configured to receive an input file, wherein the input file received may be of any file type and wherein the input file includes an associated filename;

a naming module configured to create and name an output file, wherein the output filename is generated from the associated filename of the input file and wherein the naming module receives the input file from the receiving module;

a self-extracting module configured to transform the output file into a executable file, wherein the self-extracting module receives the input file and the output file from the naming module;

Appl. No. : 09/818,134
 Filed : March 27, 2001

a loader module configured to setup the executable file to launch the input file upon execution of the executable file, wherein the loader module receives the executable file and the input file from the self-extracting module; and

a compressing module configured to compress the input file and attach the compressed input file to the executable file, wherein the compressing module receives the input file and the executable file from the loader module.

23. (Original) The system of Claim 22, wherein the loader module is further configured to setup the executable file to perform unload processes.

24. (Original) A system for creating, in response to a single action, a self-extracting file from an associated input file, wherein the associated input file is automatically launched upon execution of the self-extracting file, and wherein a user is not required to separately choose a data compression method, create a compressed archive using the chosen compression method, select an input file to be launched upon decompression of the compressed archive, and create a self-extracting file from the compressed archive, the system comprising:

means for receiving an input file to be compressed, wherein the input file may be of any file type;

means for compressing the received input file according to a data compression method; and

means for creating, in response to only a single action by a user, an executable file from the compressed input file, wherein the compressed input file will be automatically decompressed and launched upon execution of the executable file.

25. (Currently Amended) A data format, stored in a computer readable medium, comprising:

a compressed input data portion including data compressed according to a preselected data compression method;

an archive header portion, wherein the archive header portion includes information about the compressed input data portion; and

a self-extracting stub portion, wherein the self-extracting stub portion is automatically attached to the compressed input data portion and the archive header portion, and wherein the self-extracting stub portion includes a decompression engine to

Appl. No. : 09/818,134
 Filed : March 27, 2001

decompress the compressed input data portion and a loader to launch the decompressed input data portion.

26. (Currently Amended) A method for creating, in response to a single action, a self-extracting file, the method comprising:

step for receiving an input file to be used in creating a self-extracting file, wherein the input file is of any file type; and

step for automatically creating a self-extracting file using the received input file.

27. (Original) A method for creating an executable file, the method comprising:

receiving, in response to a single action, an input file to be used in creating an executable file, wherein the input file may be of any file type; and

without further instruction, creating an executable file using the received input file, wherein the executable file includes a compressed copy of the input file, and wherein the compressed copy of the input file is automatically decompressed and launched upon execution of the executable file.

28. (Currently Amended) A process for producing, in response to a single action, a computer file, the process comprising:

receiving an input file;

automatically opening an output file;

automatically adding a decompression engine to the output file for decompressing compressed data;

automatically adding loader code to the output file for launching the input file with the appropriate application software for handling the input file;

automatically adding an archive header to the output file, wherein the archive header includes information relating to the input file;

automatically compressing the input file according to a data compression method;

automatically updating the archive header to include information about the compressed input file; and

automatically closing the output file.

29. (Original) The product produced by the process of claim 28.

30. (Original) A method for creating an executable file, the method comprising:

Appl. No. : 09/818,134
 Filed : March 27, 2001

in response to a single action, receiving an input file to be used in creating an executable file, wherein the input file may be of any file type; and

without further instruction, creating an executable file using the received input file, wherein the executable file comprises:

a compressed input data portion including data compressed according to a data compression method;

an archive header portion including information about the compressed input data portion; and

a stub portion, wherein the stub portion is automatically attached to the compressed input data portion and the archive header portion, and wherein the stub portion includes a decompression engine to decompress the compressed input data portion and a loader to launch the decompressed input data portion.

31. (Currently Amended) A method for using an executable file, the method comprising:

in response to a first action, creating an executable file from any input file, wherein the executable file includes a compressed copy of the input file, and wherein the executable file includes code to decompress and to load the compressed input file; and

in response to a second action, executing the executable file to decompress; ~~wherein the compressed copy of the input file is decompressed, and launching wherein the decompressed input file is launched with an appropriate application software.~~

32. (Currently Amended) A method for creating a self-extracting file, the method comprising:

receiving, in response to a single action, an input file to be used in creating a self-extracting file; [[and]]

without further instruction, creating a self-extracting file using the input file[.]] and automatically loading ~~wherein the input file is automatically loaded upon execution of the self-extracting file.~~

33. (Original) The method of Claim 32, wherein the input file is an executable routine and wherein a function of the executable routine is called upon loading the executable routine.

Appl. No. : 09/818,134
Filed : March 27, 2001

34. (Original) The method of Claim 32, wherein the input file is a dynamic link library file.